REMARKS

Status of Claims

Claims 1-33 were previously pending, with claims 1, 6, 9, 14, 19, 24, and 29 being independent. Claims 1, 2, 6, 9, 10, 14, 15, 19, 20, 24, 25, 29, and 30 are currently amended. Thus, claims 1-33 are currently pending, with claims 1, 6, 9, 14, 19, 24, and 29 being independent.

Office Action

In the January 7, 2008, Office Action, the Examiner rejected claims 1-3, 6, 9-11, 14-16, 19-21, 24-26, and 29-31 under 35 USC 102(b) as being anticipated by Raghu et al. ("The CORAL Deductive System"), and rejected claims 4, 7, 12, 17, 22, 27, and 32 under 35 USC 103(a) as being unpatentable over Raghu et al. in view of Simon Peyton Jones et al. ("Bridging the gulf: a common intermediate language for ML and Haskell").

Summary of 2-25-2008 interview with Examiner Brent Stace

In a February 25, 2008 phone interview with Examiner Brent Stace (filling in for Examiner Cindy Nguyen), Jaclyn Alcantara (Reg. No. 61,638), presented arguments explaining why Raghu does not teach "receiving queries in a query language" as recited in all of the independent claims of the present application. Examiner Stace disagreed and recommended amending the claims to overcome the rejection. Therefore, to further distinguish the claims over Raghu, the independent claims have been amended to recite "...receiving queries in SQL..."

The Rejection of Claims under 35 USC 102

The rejections of all independent claims under 35 USC 102 rely primarily upon the teachings of Raghu to establish the existence of the claimed subject matter in the prior art. Specifically the Examiner contends that Raghu discloses (a) receiving queries in a query language the queries comprising a plurality of query terms, (b) interpreting the queries by associating at least one declarative language function with the query terms, (c) converting the queries represented by the at

12

least one declarative language function to a plurality of imperative language statements; and (d) executing the imperative language statements. Applicant respectfully submits that Raghu's "The CORAL Deductive System" fails to disclose or suggest various claimed features of the present invention, including receiving queries in a query language and interpreting the queries by associating declarative language functions with the queries.

Regarding (a) "...receiving queries in a query language..."

As amended, all independent claims of the present invention generally recite the feature of "receiving queries in SQL." The Examiner contends that Raghu, page 162, first paragraph generally discloses this feature based on the statement "The CORAL declarative language significantly extends the expressiveness of standard database query languages such as SQL..." However, this statement does not disclose or suggest receiving queries in a query language, much less a widely adopted query language such as SQL, but rather refers to an advantage of the CORAL declarative language over standard database query languages such as SQL.

Indeed, the sentence preceding the above referenced quote from page 162, first paragraph, states, "CORAL combines _features_ of database query languages (e.g., efficient treatment of large relations, aggregate operations and declarative semantics) with features of a logic programming language..." (emphasis added). Therefore Raghu teaches a system that has features of a database query language, but does not disclose receiving queries _in_ a query language with a plurality of query terms, nor does it disclose receiving queries in SQL.

This is further supported on page 161 in the "Abstract", which recites, "users can program in a combination of declarative CORAL and C++ extended with CORAL primitives." The languages disclosed here for programming in the CORAL deductive system do not include a query language and more specifically do not include SQL.

Furthermore, while Figure 3 on page 172 illustrates the "Query Evaluation System" receiving a "query", it does not disclose the query being received in a query language having a plurality of query terms, much less receiving queries in SQL. Indeed, as recited in the "Abstract", the two

13

languages disclosed in which the user can program are declarative CORAL and C++. Since the query in Fig. 3 comes from the "User Interface", the user inputs the query. Therefore the query would be sent to the "Query Evaluation System" in declarative CORAL and/or C++, not a query language such as SQL.

The Examiner also asserts that page 173, last paragraph, discloses "receiving queries in a query language, the queries comprising a plurality of query terms" The Applicant respectfully disagrees. Page 173, last paragraph, explains that the "query evaluation system" takes annotated declarative programs (in an internal representation) and database relations as input from the "Query Optimizer". It does not disclose "receiving queries in a query language, the queries comprising a plurality of query terms," nor does it disclose receiving queries in SQL. Indeed, the proceeding paragraph 3 on page 173 discloses that "complex queries typically are defined in declarative 'program modules'..." Therefore, Raghu teaches queries defined in a declarative language, not a query language such as SQL.

However, in order to avoid an overly-broad interpretation of "a query language", the Applicant has amended the claims to recite "...receiving queries in SQL...", as supported in the specification in paragraphs 41, 50, 52, etc. Note that the Applicant is not conceding that the subject matter encompassed by the claims as presented prior to this amendment is not patentable over the art cited by the Examiner. The independent claims herein were amended in this Amendment solely to facilitate expeditious prosecution of the patent application. Applicant respectfully reserves the right to pursue additional claims, including the subject matter encompassed by the claims as presented prior to this Amendment, in one or more continuing applications.

Regarding (b) "...interpreting the queries by associating at least one declarative language function with the query terms..."

Page 162, paragraphs 1-3 do not teach "interpreting the queries by associating at least one declarative language function with the query terms," as asserted by the Examiner. In particular, the Examiner recites a portion of the first sentence of the third paragraph of page 162: "Queries written

14

in a declarative language..." This quote actually teaches away from interpreting the queries by associating at least one declarative language function with the query terms, because the queries of claim 1 of the present application are received in a <u>query language</u>, not written in a <u>declarative language</u>. Therefore, Raghu does not teach or suggest interpreting the queries (received in a query language) by associating at least one declarative language function with the query terms.

Regarding Claim 2

Regarding the 102 rejection of claim 2 of the present application, the Examiner asserts that the claim is anticipated by the text found in the last two paragraphs of page 173 of Raghu. The Examiner recites, "*in addition to performing source-to-source transformations, the rewritten program is stored in a text file and also is converted into an internal representation that is used by the query evaluation system, the query evaluation system takes as input annotated declarative program (in an internal representation)...<u>the system interprets the internal form and compiled version of CORAL in which a C++ program was generated</u>.*"

First, as a note of clarification, in the quotation above, the Examiner misquotes Raghu, combining two sentences into one and leaving out key portions of each. The underlined portion of the quotation above should recite "*...system interprets the internal form of the optimized program. We also developed a fully compiled version of CORAL, in which a C++ program was generated from each user program.*" The Applicant respectfully asserts that the development of a fully compiled version of CORAL in which a C++ program was generated from each user program does not teach any of the claimed aspects of the present invention.

Additionally, since Raghu discloses queries written in a <u>declarative language</u>, not a <u>query language</u>, then Raghu does not disclose "converting the query language to an intermediate tree representation". Furthermore, "an intermediate tree representation corresponding to the at least one declarative language function" is not disclosed in Raghu. Nor does Raghu disclose converting the query to at least one data structure that is interpreted by an imperative language interpreter core to perform the queries."

However, claim 2 has been amended to reflect the amendment of claim 1. Specifically, claim 2 now recites "The method of claim 1 comprising converting the SQL to an intermediate tree representation ..."

The Rejection of Claims under 35 USC 103

The rejections of dependent claims 4, 7, 12, 17, 22, 27, and 32 under 35 USC 103 rely primarily upon the teachings of Raghu in view of Simon Peyton Jones et al. (hereafter Simon) to assert the existence of the claimed subject matter in the prior art. In light of the amendments made to the independent claims, as described above, these dependent claims are also in condition for allowance. Additionally, the Applicant respectfully disagrees with the Examiner's assertion that Simon discloses "wherein the declarative language is chosen from the group consisting of ML, LISP, and HASKELL."

Simon does not teach any of the elements claimed in the present application. The Abstract of Simon recites, "Compilers for ML and Haskell use intermediate languages that incorporate deeply-embedded assumptions about order of evaluation and side effects." However, it does not disclose a "declarative language" chosen from the group consisting of ML, LISP, and Haskell. Indeed, the phrase "declarative language" is not mentioned in Simon at all, and Simon does **not** teach that ML and Haskell are declarative languages.

Furthermore, the motivation offered by the Examiner for combining Simon and Raghu was unclear to the Applicant. The motivation provided by the Examiner on page 6 of the OA recited "to provide the ability to compile as good code as a more direct route turned out to be and identify two alternative language designs and explore the choices they embody." While this language is taken from the Abstract of Simon, it does not make grammatical sense in this context. The Applicant respectfully requests that the Examiner clarify this statement or withdraw the rejection.
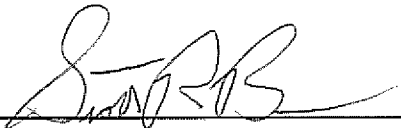
16

Conclusion

The Examiner's cited references, Raghu and Simon, alone or in combination, fail to disclose or suggest all claimed features of the present invention. Specifically, Raghu discloses a deductive system that supports a declarative language and an interface to C++, which allows for combination of declarative and imperative programming, but does not interpret, convert, or otherwise represent query statements received in SQL, as recited in the amended claims. Accordingly, Applicants respectfully request that a timely Notice of Allowance be issued in this case.

Should the Examiner have any questions, please contact the undersigned at (800) 445-3460. While the undersigned does not believe any additional fees are due in connection with this application, the Commissioner is hereby authorized to charge any additional fees associated with this communication or credit any overpayment to Deposit Account No. 09-0460.

Respectfully submitted,

HOVEY WILLIAMS LLP

By:

Scott R. Brown, Reg No. 40,535
HOVEY WILLIAMS LLP
2405 Grand Boulevard, Suite 400
Kansas City, Missouri 64108
(816) 474-9050

ATTORNEYS FOR APPLICANTS